A common problem in IoT systems is the large number of combinations of hardware, operational, and software configurations that require testing to ensure the IoT systems are free of bugs.

Due to the limitations of time and cost, there is a need for testing efforts that are both effective and efficient. The Firefly algorithm is used to minimize the test cases list.

The result of FCS is compared with Genetic Algorithm (GA), Ant Colony Optimization Algorithm (ACO), and Tabu Search (TS). It shows that FCS generates a nearly optimum result with an extremely short time.

**PRODUCT OVERVIEW**

The Firefly Combinatorial Test List Generator for IoT Systems is a software tool designed to generate test cases for IoT systems. It is designed to improve software testing practices with lesser test size, hence reducing testing costs. The tool is effective, affordable, and portable, making it suitable for use in various environments.

**COMMERCIALIZATION**

- **Price**: The first IoT combinatorial testing strategy based on Modified Greedy Algorithm. Sustainability practices and contributions improve software testing practices with lesser test size, hence reduce testing costs.
- **Effective**: It can be used in any platform environment such as OS, Linux, and Windows.

**MARKETABILITY**

- Software prototype for Testing & Maintenance

**ACHIEVEMENTS**

- An experimental study of hyper-heuristic selection and acceptance mechanism for combinatorial t-way test suite generation. Information Sciences, WebofScience, JCR, Q1
- Gold Medal at CITREX2018
- Gold Medal at BIS2017

**FONDS**

- FRGS "Input-Output Relations Harmony Search T-way Testing Strategy" RM90K
- FRGS "A Reinforcement Learning Sine Cosine Based Strategy for Combinatorial Test Suite Generation" RM60K
- International Research Fund RM200K King Khalid University
- University Malaysia Pahang Research Funds RM80K
- Received RM90K from King Khalid University for

**PATENT/COPYRIGHT**

- Applied for the Patent

**NOVELTY & INVENTIVENESS**

- The first IoT combinatorial testing strategy based on Modified Greedy Algorithm.
- Sustainability practices and contributions improve software testing practices with lesser test size, hence reduce testing costs.
- More effective at finding software faults.
- Result of Comparing with Existing Strategies, it can be used in any platform environment such as OS, Linux, and Windows.